社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－ＡＲＣ－168（12）
2006／6／9

高安全自動車道路抽出のための動的再構成可能アーキテクチャ

李 承啓† 張山 昌論†† 亀山 充隆††

東北大学大学院情報科学研究科 〒980-8579 宮城県仙台市青葉区荒巻字青葉 6-6-05
E-mail: †lsg@kameyama.ecei.tohoku.ac.jp, ††{hariyama,kameyama}@ecei.tohoku.ac.jp

あらまし 本論文では，高安全知能自動車実現のための道路を抽出するアルゴリズムとその専用ハードウェア化について提案する．左右のステレオ画像を用いることで，道路が平面である点に着眼し，その幾何学的特徴を表す射影変換行列を利用することで道路領域を抽出する．左右のステレオ画像の一方を射影変換し，もう一方の画像と差分画像及びSAD(Sum of Absolute Difference) を用いた検証をすることで道路を抽出する．基本処理である射影変換及び検証による道路領域抽出は計算量が多いため，専用ハードウェア化を行う．射影変換行列の推定では，主に行列変換とソーティング処理が行われる．検証による道路領域抽出においては画素の輝度値の差分やSADを用いる．これらは，全て加算ベースの動的再構成アーキテクチャを提案する．画像サイズ 720×480 の場合を考慮してみると，一般のハードウェアでは約2.5秒の時間を要したが，提案されたハードウェアでは約0.05秒以下の時間で処理が可能であると予測される．

キーワード 射影変換，動的再構成プロセッサ，絶対値差分和

Processor Architecture for Road Extraction Based on Projective Transformation

Sunggae LEE†, Masanori HARIYAMA††, and Michitaka KAMEYAMA††

Graduate School of Information Sciences, Tohoku University 6-6-05 Aoba, Aramakiazana, Aoba, Sendai, Miyagi 980-8579 Japan
E-mail: †lsg@kameyama.ecei.tohoku.ac.jp, ††{hariyama,kameyama}@ecei.tohoku.ac.jp

Abstract This paper presents a processor architecture for a robust road-extraction algorithm using stereo images. The road-extraction method estimates a projective-transformation matrix H that represents the geometrical features of road regions. The road region can be robustly extracted using Sum of Absolute Differences (SAD) between one image and a transformed image of the other one. To accelerate the projective transformation, dedicated multipliers are employed. A dynamically reconfigurable architecture for robust estimation and SAD-based extraction is presented. For image size 720×480, the execution time on the proposed processor is 0.05 seconds whereas that on a microprocessor is 2.5 seconds.

Key words projective transformation, homography, dynamically reconfigurable processor, SAD

1. まえがき

道路領域の正確な抽出は高安全自動車において，基本となるものである．高安全知能自動車において，視覚による走行領域の認識は移動車を走行路に沿って，かつ障害物を避けながら誘導するための重要な技術の１つであり，従来からさまざまな手法が提案されている．

代表的な方法として，テクスチャ情報に基づく処理が存在する．例えば，路面上の白線を抽出し，その間を道路と認識する手法がある．また，道路の境界のエッジを抽出することで道路を検出する手法や，道路の消失点の位置と方向を用いることで道路領域を抽出する手法も存在する．しかし，これらの手法は，主に道路の特定パターンや形状的特徴に依存しているため，限定されたシーンのみに有効という問題点がある．これに対して，シーンの三次元的情報を用いることで道路領域を抽出する手法も提案されている．一般には，ステレオ画像から距離情報を求め，この情報から走行領域を認識する手法が用いられるが，この手法は，距離情報を求める難易性から道路領域を抽出するのは容易ではない．

そこで，全体の距離情報を求めることなく、道路が平面で

あることに着目し，平面の幾何学的特徴を表す射影変換による変換画像と基準画像をマッチングする手法が提案されている [1], [2]．この手法は，道路抽出の品質に影響を与える重要な処理であり，SAD(Sum of Absolute Difference) を用いた対応点探索による方法 [1] や差分画像処理による方法 [2] がある．しかしながら，SAD を用いた対応点探索による方法は計算量，差分画像処理による方法は信頼性に問題がある．そこで，2 つの手法を組み合わせることで，これらの問題点を解決するアルゴリズムを提案する．

また，これらの方法は，計算量が多いためリアルタイムに応用するには困難である．全体の処理を分析し，最も計算時間を要する部分を求め，この処理部分の専用ハードウェア化を図る．主に，射影変換行列の推定と検証による道路領域抽出の処理が全体の 9 割を占める．射影変換行列の推定は，さらに積和演算による行列計算とその計算結果のソーティングに分けられる．また，検証による道路領域抽出は，そのテクスチャ的特長によって差分と SAD の方法に分けて処理する．これらの処理はどれも加算処理が基本であるため，加算器をベースにすることで演算部を構成することが可能となる．この共通部を利用することで各処理によって別々のハードウェアを用いるのではなく処理によって機能を切り替えることができる動的再構成可能なハードウェア化を目的とする．

2. 道路領域抽出

2.1 基本概念

本アルゴリズムは，道路が平面であることに着目し道路を抽出する手法である．3 次元空間中の点群が平面上に存在するとき，一方の画像の点と他方の画像に存在する対応点は射影変換行列 H より次の式が成り立つ [3]．

$$m_B = Hm_R \tag{1}$$

ここで，m_B，m_R はそれぞれ基準画像と参照画像の座標であり，対応点関係にある．三次元上に存在する点を 2 台のカメラを用いて異なる画像に投影したとき，その点どうしは幾何学的に同一の点であり，この 2 つの点を対応点と呼ぶ．その関係を図 1 に示す．

ここで，M が三次元上に存在する点であり，m_1，m_2 がそれぞれ左右の画像に投影された点であると同時に対応点である．式 (1) により，道路平面上に存在する点の対応点どうしは射影変換を行うことで座標が完全に一致し，それ以外では一致しない．例として，図 2(a) の基準画像と図 2(b) の参照画像を用いる．参照画像を射影変換した画像を図 2(c) として，道路領域を抽出した結果を図 2(d) とする．B1 と R1，B2 と R2 はそれぞれ対応点どうしであり，B1 と R1 は道路平面上に存在する点，B2 と R2 は道路平面上以外に存在する点である．T1 と T2 は R1 と R2 を射影変換した結果である．このとき，対応点どうしである B1 と T1 は道路平面上の点であるため式 (1) が成立する．一方で，B2 と T2 は対応点どうしではあるが，道路平面上の点ではないため式 (1) が成立しない．

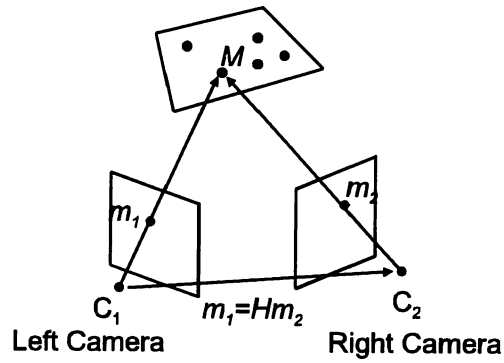

図 1 三次元的位置関係

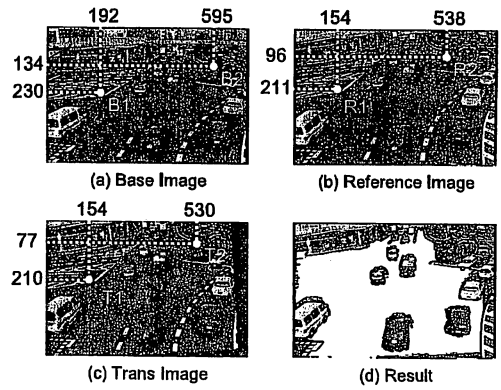

図 2 道路領域抽出

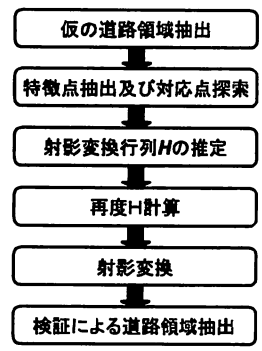

図 3 処理のフローチャート

2.2 道路抽出アルゴリズム

処理の全体の流れを図 3 に示す．この道路抽出処理の中で要となる処理は，正しい射影変換行列 H の推定と検証による道路領域抽出の 2 つであり，これらに焦点をおいて論じる．まず，正しい射影変換行列の推定を説明する．画像に存在する対応点を 4 つ選ぶことで射影変換行列を求めることが可能となる．しかし，画像上には道路平面以外にも存在する点が存在するため，このような点を例外点として除去しないと正しい射影変換行列

を求めることは出来ない．そのため，対応点を 4 つ選び射影変換行列を求め，それを用いて射影変換した結果をロバストに推定する [1] ことで正しい射影変換行列を求める．このとき，正しい射影変換行列の推定は，さらに射影変換の計算 (行列計算) のための積和演算と演算結果のソーティングの 2 つの処理を行うことで結果を求める．式 (2) に行列計算を示す．

$$DS = \left(\begin{pmatrix} m_{Bx} \\ m_{By} \\ 1 \end{pmatrix} - \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} m_{Rx} \\ m_{Ry} \\ 1 \end{pmatrix} \right)^2 \quad (2)$$

m_{Bx}, m_{By} は基準画像の x，y 座標であり，m_{Rx}, m_{Ry} は基準画像の x，y 座標である．また，この計算においては全て同次座標系での計算である．DS(Difference-Square) のソーティングを行うことで，行列計算を用いる H の正誤を推定する．画像上でもっとも大きな部分を示すものが道路と仮定するため，同じくその部分に含まれる対応点が最も多く，画像上からランダムに対応点を求めたとき，道路に存在する点の DS は道路に対応する H であればほぼ 0 に近い値を示す．よってソーティングを行うことで最も多くより 0 に近い値を示す H を推定する．次に，検証による道路領域抽出である．射影変換行列が正しく導出されたとき，参照画像を射影変換すれば道路平面に対応する点は基準画像の対応点と同一の座標に変換され，それ以外は異なる座標に変換される．この性質に着目し，基準画像と変換画像の間で同一座標間の画素値の差分を求め，その値が十分に小さければ道路であると判定する．ここで，注意するべきことは，対応点関係は基準画像と参照画像間に成り立つ関係であるが，画素値の比較は基準画像と参照画像を変換した変換画像間で行うという点である．差分のみで道路を抽出したとき，射影変換行列の精度の問題 (対応点が誤った対応や道路が完全に平面ではないことが原因となる) から道路平面であっても白線のような境界部分では変換後の画像が完全に一致しない場合がある．すなわち，式 (1) が成立しない場合がある．これを解決するため，境界部分では SAD を用い変換された座標付近を探索する．例えば，基準画像の図 2(a) の点 B1 の対応点を変換画像の図 2(c) の点 T1 の周辺を SAD を用いて探索し，その値が一定値以下なら道路と判定する．このように，ある閾値を設定して検証をすることで少しのずれがあっても道路抽出が可能となる．

2.3 道路領域抽出の問題点

実行結果を図 2(d) に示す．この処理には，画像サイズ 720 × 480 を用いたとき，Intel Pentium M(1GHz) のコンピュータで約 2.5 秒を要した．提案手法では，SAD と差分画像を用いる道路領域検証を行うことで，信頼性を損なわずに高速化が可能となった．しかしながら，処理時間には約 3 秒弱が要するため，リアルワールドに応用するにはさらなる高速化が必要となる．表 1 に各処理の計算時間を示す．特に問題となる処理は，射影変換行列 H の推定と検証による道路領域抽出が全体の 90 %以上であることが確認できる．この 2 つの処理を高速化するために，専用のハードウェア化を目標とする．

表 1 各処理の計算時間

処理名	計算時間
仮の道路領域抽出	0.0250
特徴点抽出及び対応点探索	0.04891
射影変換行列 H の推定	1.2552
再度 H 計算	0.0002
射影変換	0.0281
検証による道路領域抽出	1.4899

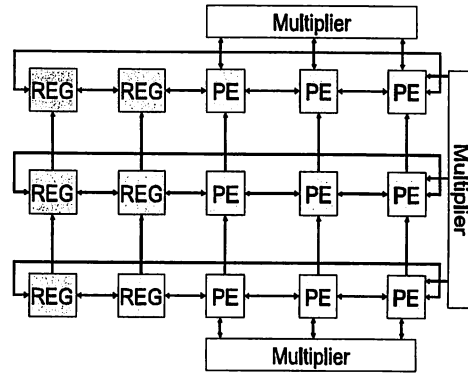

図 4 全体の回路図

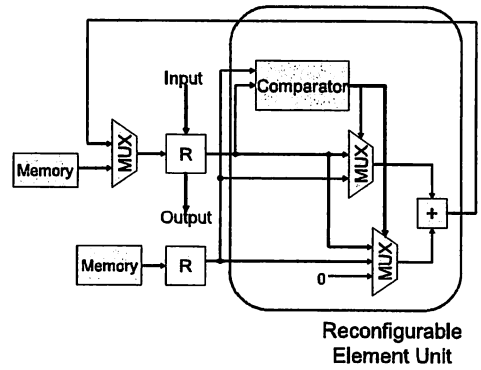

図 5 Processing Element

3. 専用のハードウェア化

3.1 全体の構成

処理の要である正しい射影変換行列の推定と検証による道路領域抽出の処理時間が全体の 9 割以上を示す．そこで 2 つの処理を専用ハードウェア化することで高速化を図る．全体の回路を PE(Processing Element) アレイが 3×3 のときの例として図 4 に示す．PE は，基本演算部である．2 つの処理は加算器をベースとして構成されている．この共通部分を PE として機能を切り替えることで，専用ハードウェアを 2 つ用いるのではなく 1 つの回路で実現する [4]．図 5 に PE を示す．PE の REU(Reconfigurable Excution Unit) が，処理によってその機能を切り替える部分となる．

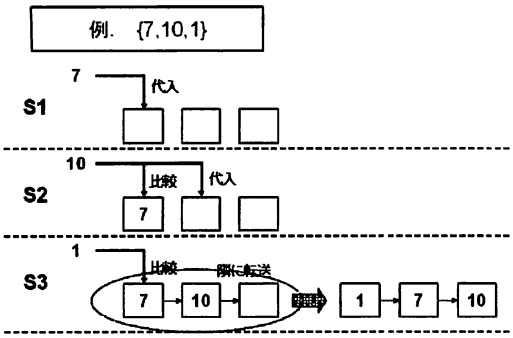

図 6 ソーティングの例

3.2 射影変換行列の推定処理の高速化

問題となる正しい射影変換行列の推定は次のようになる．射影変換行列 H の推定は，行列計算 (積和演算) とソーティングの 2 つに大きく分けられる．積和演算は，乗算部分があるために，計算量が多くなるため，図 4 に示したように専用の乗算器 (Multiplier) を組み込むことで高速化を図る．これによって専用の乗算器の数性能によって高速化が可能となる．ソーティングについては，図 6 に例を示す．上図にあるように，データを保存し新データが入力されたときにはその値を比較することで元のデータを書き換えるか，そのままにするか，隣に転送するかを決める．全データに対して一度に処理することで高速にソーティングが行われる．この比較転送を行う部分を PE とすれば，ソーティングはいくつもの PE で並列に処理をする．最大約 100 個までのデータを一度に計算する必要があるが，ハードウェアリソース面から考えても PE は 100 個以上を組み込むため，ソーティング処理は非常に高速に処理できる．

3.3 検証による道路領域抽出処理の高速化

次に，検証による道路領域抽出は，差分と SAD の 2 つを用いる．差分は，画像間の画素値の減算のみなので高速に計算ができる．しかしながら，SAD 処理は画像上で探索が必要となるために計算時間が長い．そのため SAD 処理は高速化を図る必要がある．SAD は 2 枚の画像間の類似度を加算のみで評価できる代表的方法であるが，一般の方法では同じデータを幾度もロードして重複計算を行うなど高速な演算には適しない [5]．しかし，図 4 のように PE と REG を配置することで解決できる．メモリは PE アレイの一列にデータを転送し，この値をシフトさせながら PE に必要な値が揃ったときに SAD の計算を行う．また，一度ロードした値はさらに REG にシフトさせ保存し，必要なときに逆方向にシフトさせることで無駄なメモリアクセスも減らすことができる．具体例を図 7 に示す．左の C1-C25 が探索範囲の画像データあり，太線で囲まれている部分が SAD 計算をする部分である．また，右のアレイは PE と REG を表しており SAD 計算部は PE の部分でありこの部分で実際の演算を行う．前提として，PE にはもう一方の画像データが入力されているとする (もう一方の画像データは探索範囲の SAD を全て計算するまで変わらない)． のときには，メモリから一列ずつ値をロードしシフトさせることで SAD 計算に

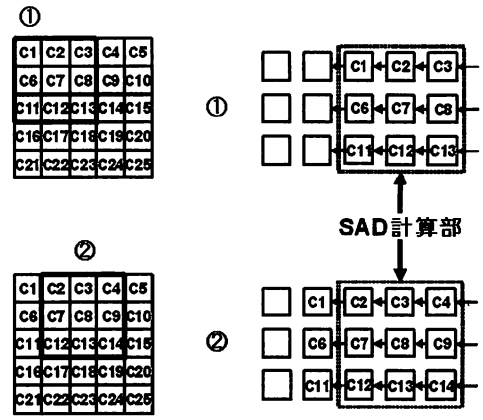

図 7 SAD 処理の例

必要な値を揃える． のときは，さらに値をシフトさせロードさせることで SAD 計算の計算に必要な値を配置させる．例からも確認できるように画像データ常に同一方向にシフトするため，この制御は比較的に楽にできる．また，SAD を計算するウィンドウサイズに比例して回路を組むために，高速かつ効率な処理が可能となる．

3.4 メモリアロケーション

メモリからのロード時間を 1 ステップとすると，K 個のデータを読み出すためには K ステップを要す．SAD 演算のような大量のデータを読み出す処理においては，処理時間のボトルネックとなる．複数のメモリモジュールを用いて必要なデータを重複記憶させることも可能だが，この方式は，配線の複雑化および記憶容量の膨大化が問題となる．したがって，メモリモジュールの数および容量が最小となるダイアゴナルメモリアロケーションを用いる [4], [5]．データは PE アレイの縦方向もしくは横方向から n 個ずつ読み出される (n は，PE の縦一列および横一列の個数)．$n = 9$ の場合の例を図 8 に示す．各画素の番号はメモリ番号に対応している．各画素が保存されるメモリ番号 $M(x, y)$ は次の式 (3) より求まる．

$$M(x, y) = (x + y) \bmod n \tag{3}$$

x, y は画像の x, y 軸に該当する．縦 9 画素及び横 9 画素のすべての読み出しパターンにおいて，画素データ 1 画素ずつ異なるメモリモジュールに保存されている．これによって，1 ステップでは 9 個の画素データを最小のメモリモジュール数 9 個で並列に読み出し可能となる．

4. 評 価

提案するハードウェアの計算量として予測される結果を表 2 に示す．全ての処理は，一般のハードウェアの計算量を 1 として比較した結果を表 2 示す．積和演算は，専用の乗算器を組むため，その個数分だけの高速化が予測される．3×3 行列の積和演算を行うため，最大 9 個の乗算を一度に行うことができ，最大 9 倍高速化が期待できる．ソーティングでは，縦一列の PE

0	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	0
2	3	4	5	6	7	8	0	1
3	4	5	6	7	8	0	1	2
4	5	6	7	8	0	1	2	3
5	6	7	8	0	1	2	3	4
6	7	8	0	1	2	3	4	5
7	8	0	1	2	3	4	5	6
8	0	1	2	3	4	5	6	7

図 8 メモリアロケーションの例 ($n = 9$)

表 2 ハードウェアの評価

	一般ハードウェア	提案ハードウェア
積和演算	1	1/9
ソーティング	1	$1/n$
SAD 検証	1	$1/n^2$

を並列に利用するため，最大 n 倍高速化が期待できる．SAD 検証では，SAD を求める画像サイズ分だけの PE を準備したハードウェアを基本としているため SAD の画像サイズに比例し，PE の全個数である最大 n^2 倍高速化が期待できる．画像サイズ 720 × 480 の場合を考慮してみると，一般のハードウェアでは約 2.5 秒の時間を要したが，提案されたハードウェアでは約 0.1 秒の時間で処理が可能であると予測される．

5. むすび

道路領域抽出のための再構成可能なプロセッサアーキテクチャを提案した．このアーキテクチャは隣接間のデータ転送を可能とするシフトレジスタが主となって構成されている簡単なインターコネクッションを持つ構造である．提案されたアーキテクチャは，フルカスタム設計のみではなく，インターコネクッションのディレイがロジックディレイより問題より問題となる FPGA にも有用なプロセッサとして様々な応用が期待できる．

文　献

[1] 奥富正敏，野口卓，中野勝之，ステレオ画像からの射影変換行列の抽出による道路領域検出，日本ロボット学会誌，Vol.18, No.8, pp.51-57, 2000.

[2] 奥富正敏，中野勝之，丸山純一，原智章，ステレオ画像を用いた視覚誘導のための平坦部の連続推定，情報処理学会論文誌，Vol.43, No.4, pp.1061 − 1069, 2002.

[3] 出口，ロボットビジョンの基礎，コロナ社，2000.

[4] 張山昌論，李昇桓，亀山充隆，転送ボトルネックのないセンサ・メモリアーキテクチャに基づくモーションステレオ VLSI プロセッサの構成，T.IEE Japan, Vol.120-E, No.5, pp.237-244, 2000.

[5] 小林康浩，張山昌論，亀山充隆，最適アロケーションに基づくオプティカルフロー抽出プロセッサの FPGA 実現，IEEJ Trans. FM, Vol.123, No.1, pp.1-11, 2003.